NORQUEST COLLEGE

CMPT1011

Introduction to Computing

Unit 4: Python Built-in Functions

Learning Outcome

- Apply function calls in your program
- Apply math function in your code
- Define parameters and arguments in a built- in function

Functions

Function

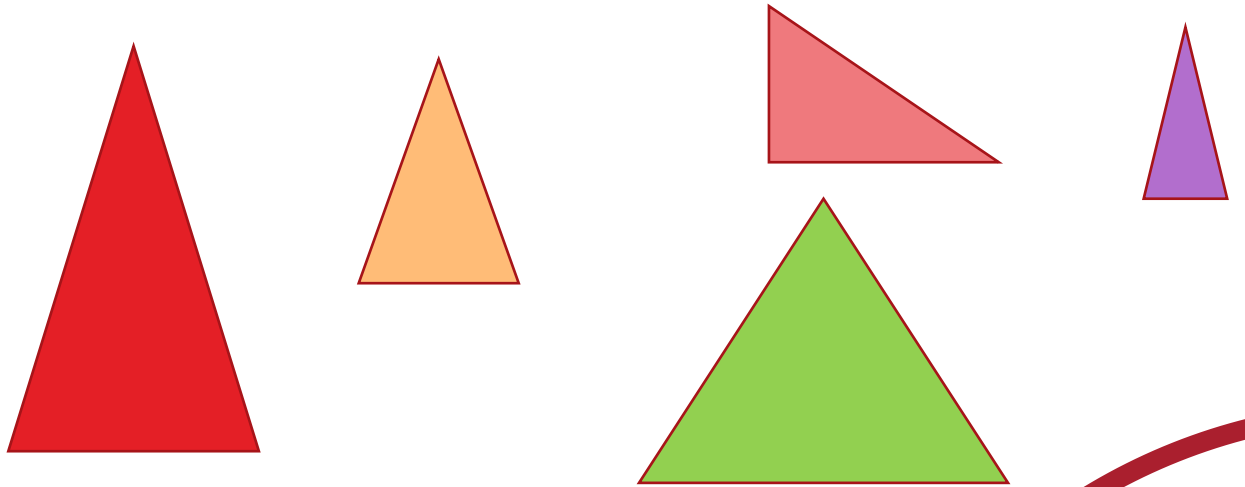
- A function is a block of code that performs a specific task.
- The main idea is to put the repeatedly performed task in a function to avoid rewriting the code for different inputs.
- We use function call to reuse the code with different input.

Function

- A function is a block of code that performs a specific task. E.g.,

- To create a triangle and color it, we can make two functions:

1. Create triangle
2. Color triangle



Function

- Dividing a complex task into smaller chunks, make it easier to understand and reuse.
- There are two types of functions in Python:

1. **Standard library functions**
2. **User-defined functions**

Standard library functions

- These are built-in functions available in Python to use. E.g.,

- input()
- type()
- print()
- Function names are followed by a pair of brackets.
- You can use or call a function by typing its name.

Program using Built-in function

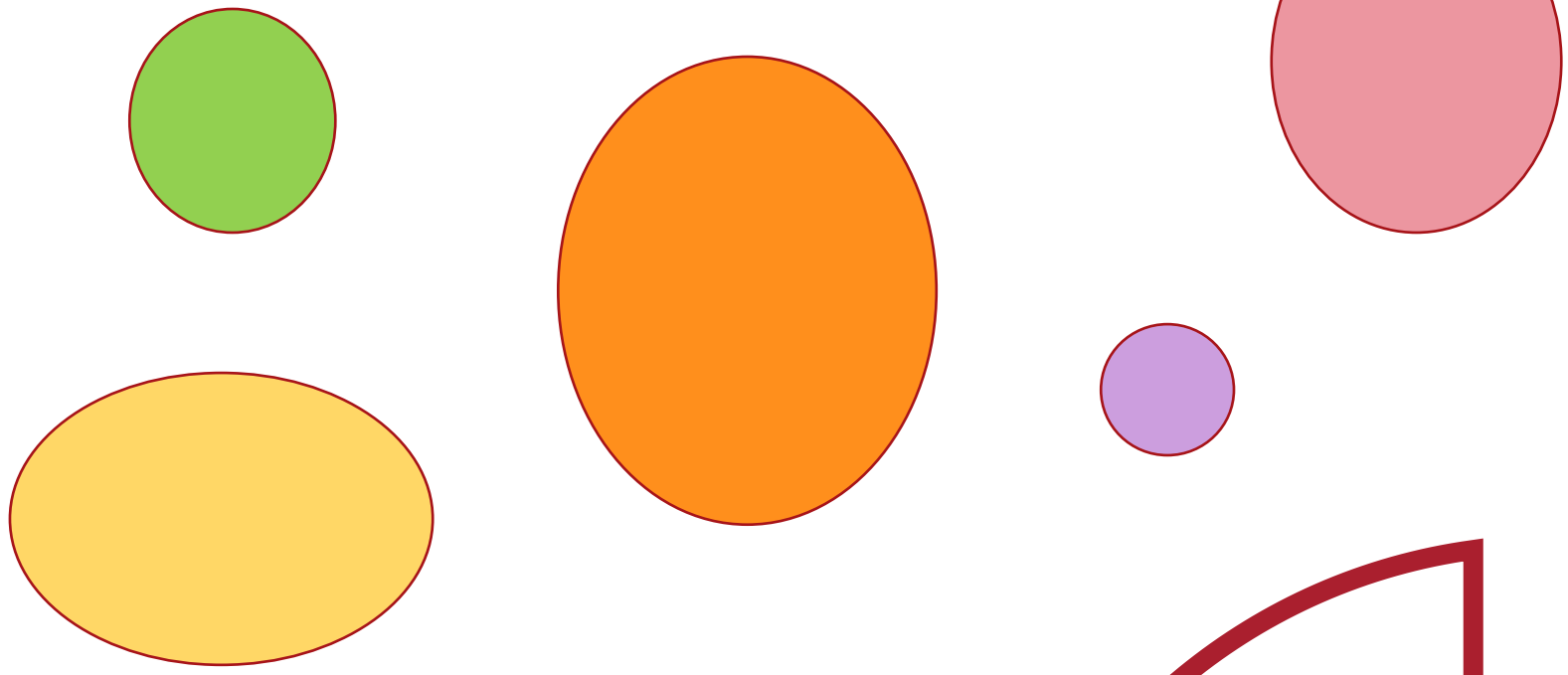
- Write a program to find square root of a number.
- Hint: the function is sqrt in math module

- import math
- math.sqrt(any number)
- math.pow(base, power) 6pow 3
- math.floor(3.6)
- math.ceil(3.6)

User-defined functions

- These are the functions created by user according to their requirements.

- Create circle
- Color circle etc.



Python function declaration

- The syntax to declare a function is:

```
def function_name(arguments):
    # function body

    return
```

Python function declaration

- **def** is used to declare a new function.
- **function_name** is used to give any user defined name to a function.
- **arguments** is any value passed to function.
- **return** (optional) value returned from a function

Python function declaration examples

```
def greet():
    print('Hello World!')
```

```
def add_numbers():
    ...
    return sum
```

```
def my_function():
  print("Hello from a function")

my_function()
```

Calling a Function

To call a function, use the function name followed by parenthesis:

for example:
Function_name(a,b)
sum()

Python function declaration examples

```
def greet():
    print('Hello World!')
```

```
# call the function
greet()
```

```
def add_numbers(a, b):
    return a + b

result = add_numbers(3, 5)
print(result)
```

Parameter

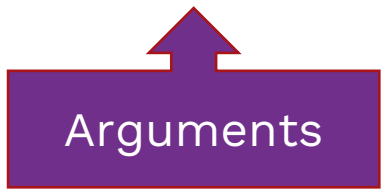
A parameter is the variable listed inside the parentheses in the function definition

for example:
function(a,b)

Argument

An argument is the value that is sent to the function when it is called

for example:
function(2,3)



Return value

A function takes an argument and “returns” a result

The result is also called the return value

Hands on

Write a program using functions to add two numbers.

Addition without return

```
def addition(num1, num2):  # Function header
    num3 = num1 + num2
    print("The sum of two numbers-", num3)


addition(4, 5)  # function calling
```

Addition with return

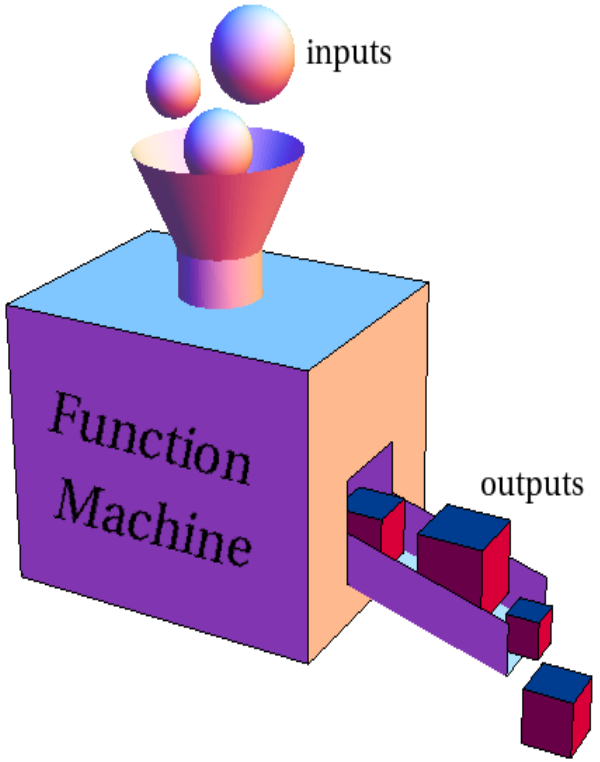
```
def addition(num1, num2):  # Function header
    num3 = num1 + num2
    return num3



answer = addition(4.5, 5)  # function calling
print("The sum of two numbers-", answer)
```

Math Functions

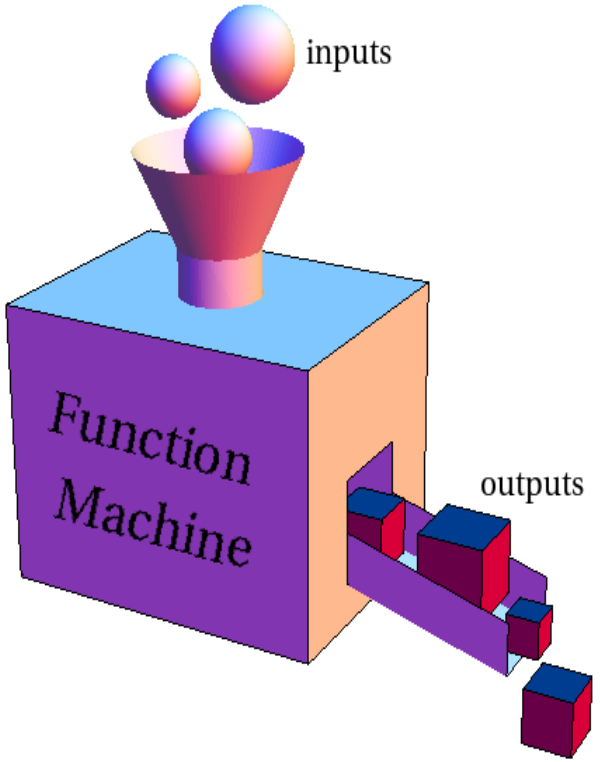
Math functions



This Photo by Unknown Author is licensed under CC BY-SA-NC

- Python has a math module that provides most of the familiar mathematical functions
- A module is a file that contains a collection of related functions

Math functions

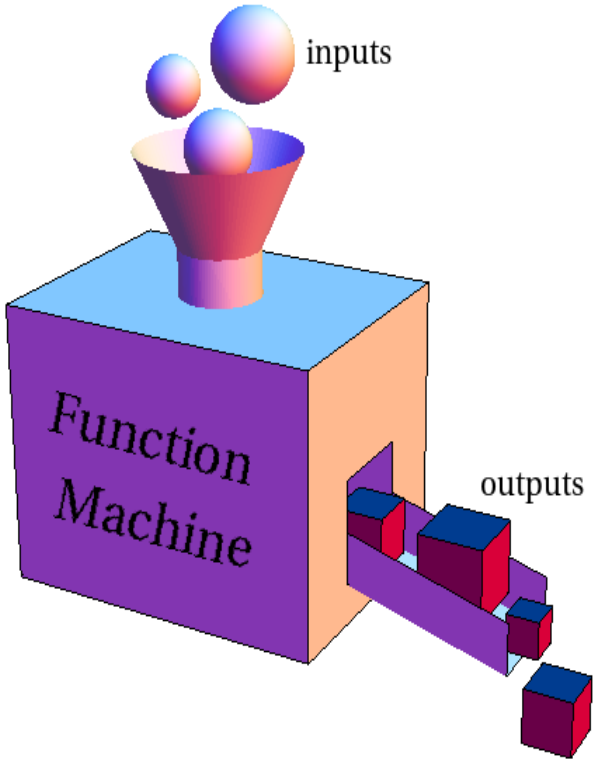


- Before we use the functions in a module, we have to import it with an import statement:

- import math

This Photo by Unknown Author is licensed under CC BY-SA-NC

Math functions



- To access one of the functions, you have to specify the name of the module and the name of the function, separated by a dot

This Photo by Unknown Author is licensed under CC BY-SA-NC

A more deeper look to “import”

- If you want to use a module, you can include it in your code by importing it e.g.

import math

- To access the functions inside math module, use dot(.) operator. E.g.,

math.sqrt(x)

A deeper look to “import”

- It is also possible to import specific function from a module. E.g.,

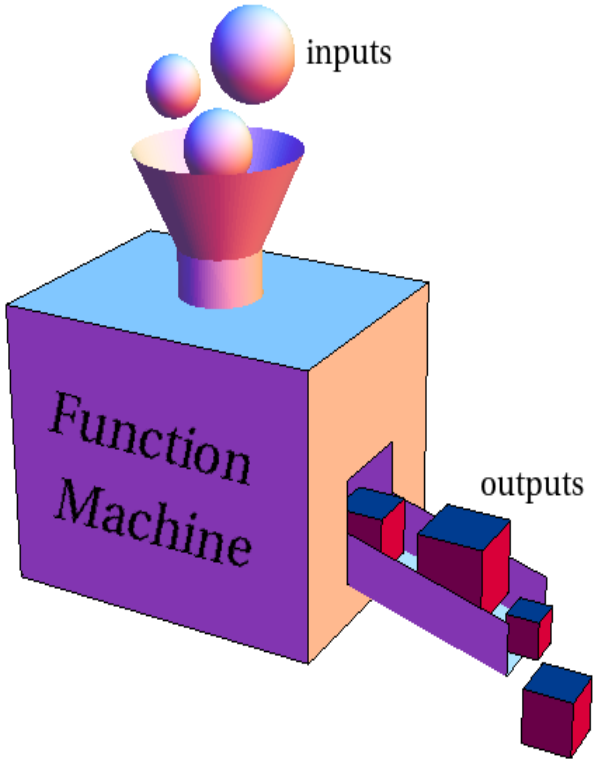
```
from math import floor, ceil
ceil(x)
floor(x)
```

- You can import all identifiers from a module using wild card import (*)

```
import math *
```

- *Wildcard import can lead to subtle errors and should be avoided*

Math functions



- https://docs.python.org/3/library/math.html

This Photo by Unknown Author is licensed under CC BY-SA-NC

Math functions

1. math.**ceil**(*x*)
2. math.**fabs**(*x*) *returns **floating** absolute value*
3. math.**factorial**(*n*) *return error if n is not integral or neg*
4. math.**floor**(*x*)
5. math.**cbrt**(*x*) *return cube root of x*
6. math.**pow**(*x*, *y*) *return x raised to power y*

Explore more functions:
https://docs.python.org/3/library/math.html

Built-in Functions

Built-in Functions

The Python interpreter has a number of functions and types built into it that are always available

https://docs.python.org/3/library/functions.html

abs(x)

Return the absolute value of a number

For example:
abs(-56)
Abs(98.6)

max(arg1, arg2, *args[, key])

Return the largest item in an iterable or the largest of two or more arguments.e.g.,

```
x = max(5, 10)
print(x)

x = max("Mike", "John", "Vicky")
print(x)
```

```
cook@pop-os:~$ ascii -d
   0 NUL     16 DLE     32          48 0     64 @     80 P     96 `    112 p
   1 SOH     17 DC1     33 !        49 1     65 A     81 Q     97 a    113 q
   2 STX     18 DC2     34 "        50 2     66 B     82 R     98 b    114 r
   3 ETX     19 DC3     35 #        51 3     67 C     83 S     99 c    115 s
   4 EOT     20 DC4     36 $        52 4     68 D     84 T    100 d    116 t
   5 ENQ     21 NAK     37 %        53 5     69 E     85 U    101 e    117 u
   6 ACK     22 SYN     38 &        54 6     70 F     86 V    102 f    118 v
   7 BEL     23 ETB     39 '        55 7     71 G     87 W    103 g    119 w
   8 BS      24 CAN     40 (        56 8     72 H     88 X    104 h    120 x
   9 HT      25 EM      41 )        57 9     73 I     89 Y    105 i    121 y
  10 LF      26 SUB     42 *        58 :     74 J     90 Z    106 j    122 z
  11 VT      27 ESC     43 +        59 ;     75 K     91 [    107 k    123 {
  12 FF      28 FS      44 ,        60 <     76 L     92 \    108 l    124 |
  13 CR      29 GS      45 -        61 =     77 M     93 ]    109 m    125 }
  14 SO      30 RS      46 .        62 >     78 N     94 ^    110 n    126 ~
  15 SI      31 US      47 /        63 ?     79 O     95 _    111 o    127 DEL
```

Ref: **John D. Cook , (28 May 2022), How to memorize the ASCII table https://www.johndcook.com/blog/2022/05/28/how-to-memorize-the-ascii-table/**

min(arg1, arg2, *args[, key])

Return the smallest item in an iterable or the smallest of two or more arguments. e.g.,

min(‘orange’)
Output: ‘a’

Hands-on practice

Practice

1. Write a program to calculate area of a circle.

Area of a circle = πr^2 $\pi = 3.1415$

Use math module for value of pi (*hint: math.pi*)

Practice

2. Write a program to find the minimum value from the list = [“yellow”, “red”, “blue”, “orange”, “green”]

Practice

3. Write a program to initialize a variable outside a function and print the value:

i. inside the function

ii. Outside the function

Questions?

References

1. https://greenteapress.com/wp/think-python-2e/
2. https://www.w3schools.com/python/python_functions.asp

NORQUEST COLLEGE